

K-Nearest Neighbors (K-NN)

Prof. André Gustavo Hochuli

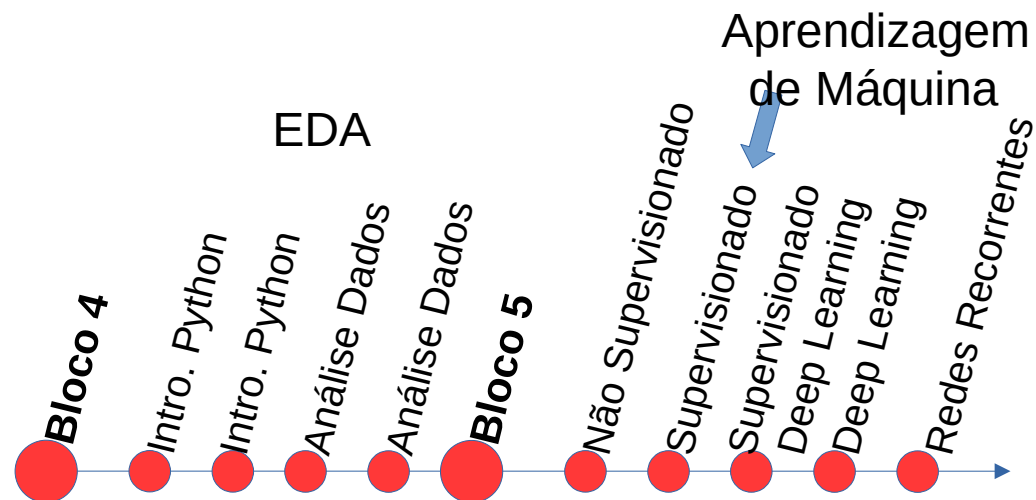
gustavo.hochuli@pucpr.br

aghochuli@ppgia.pucpr.br

github.com/andrehochuli/teaching

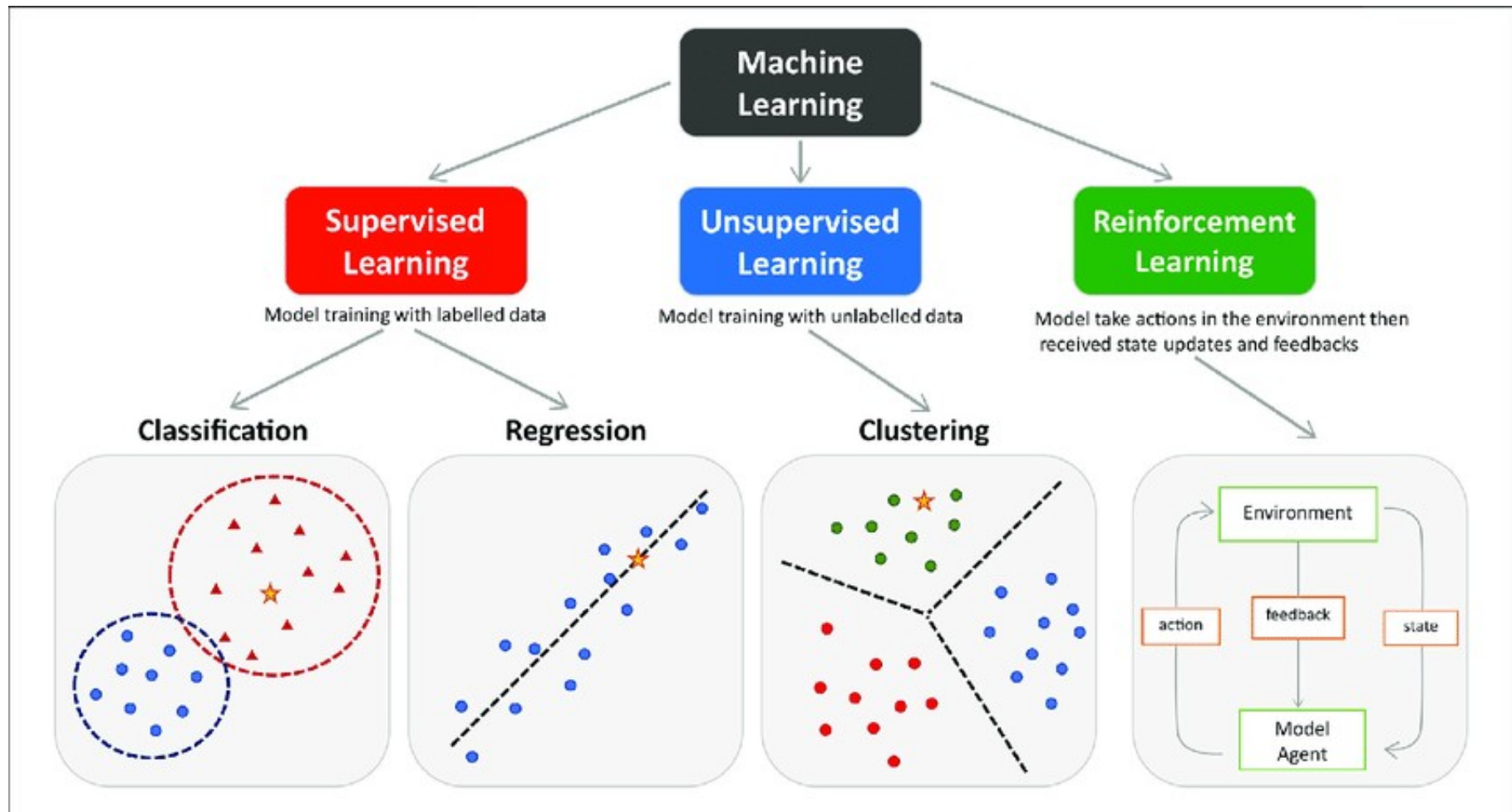
Plano de Aula

- Discussões Iniciais
- Aprendizado por Instâncias
- Algoritmo KNN
- Métricas de Avaliação
- Exercícios



Discussões Iniciais

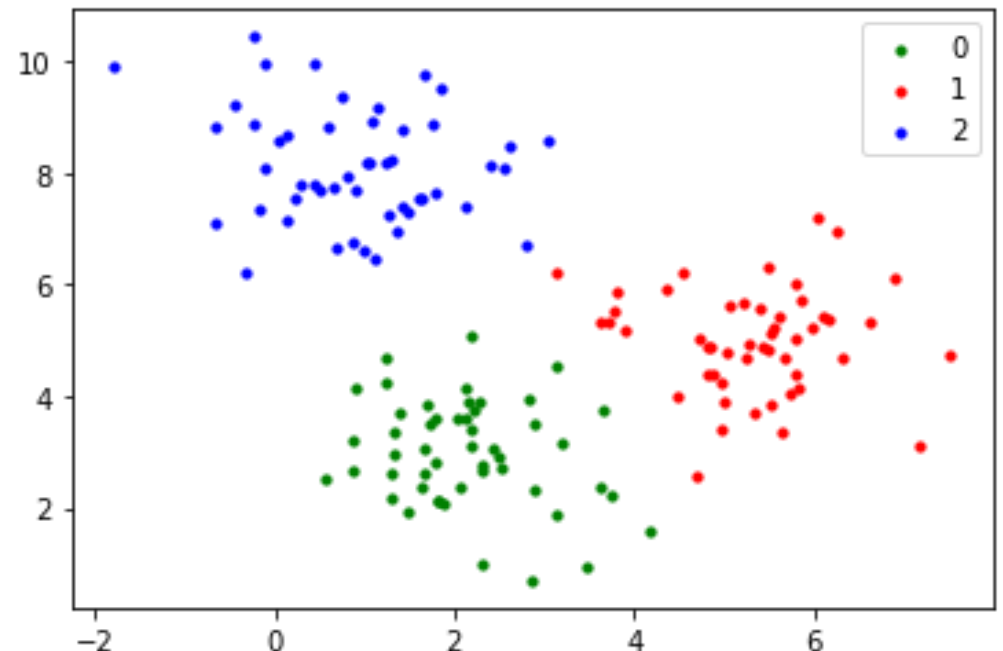
Tipos de Aprendizado



Aprendizado por instâncias

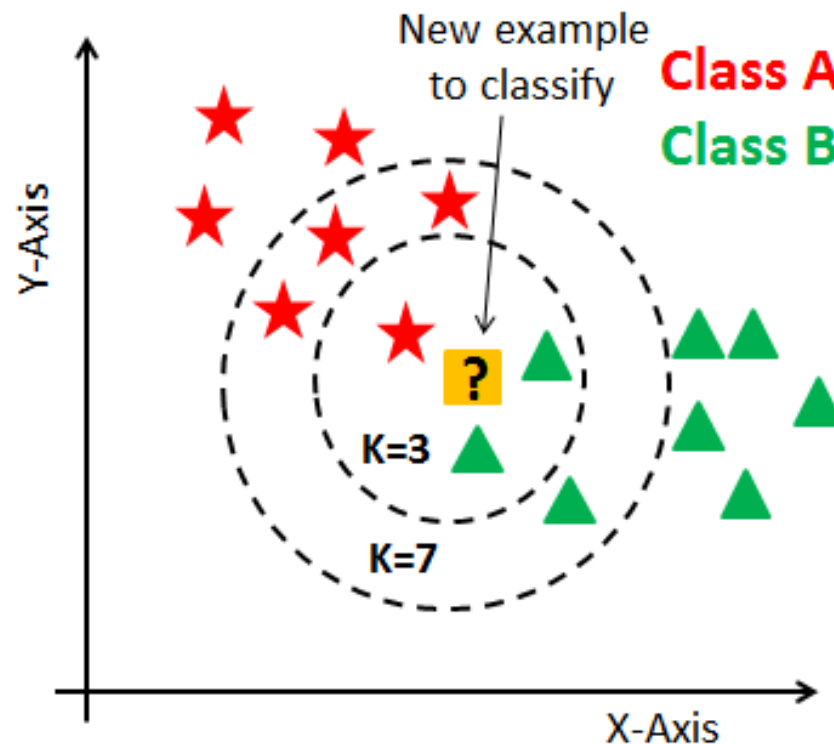
- Características são mapeadas no espaço Euclidiano
 - Métodos não paramétricos
 - Distribuições Arbitrárias
 - Sem suposição sobre as densidades

```
[ -0.23685338  8.87583893] 2
[ 1.01652757  8.17718772] 2
[ 2.55880554  8.1094027 ] 2
[ 4.86355526  4.88094581] 1
[ 6.12141771  5.40890054] 1
[ 5.04366899  4.77368576] 1
[ 2.31563495  0.97779878] 0
[ 5.84616065  4.14048406] 1
[ -0.2197444  10.44936865] 2
[ 1.6775828   2.61594565] 0
```



K-Nearest Neighbors (K-NN)

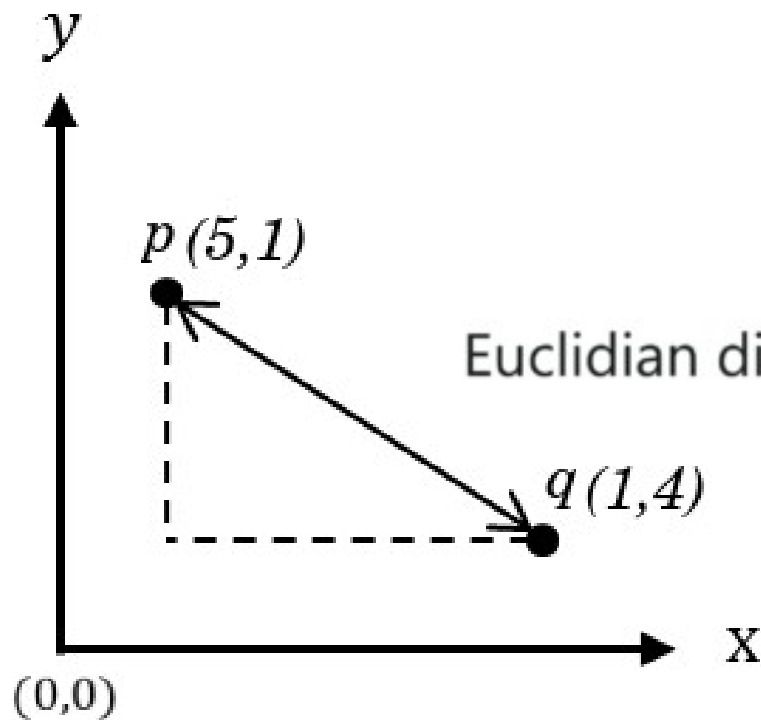
- Votação de 'K' vizinhos da amostra de teste



Distância Euclidiana

- Determina a distância entre dois pontos espaço euclidiano

$$c^2 = a^2 + b^2.$$



$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

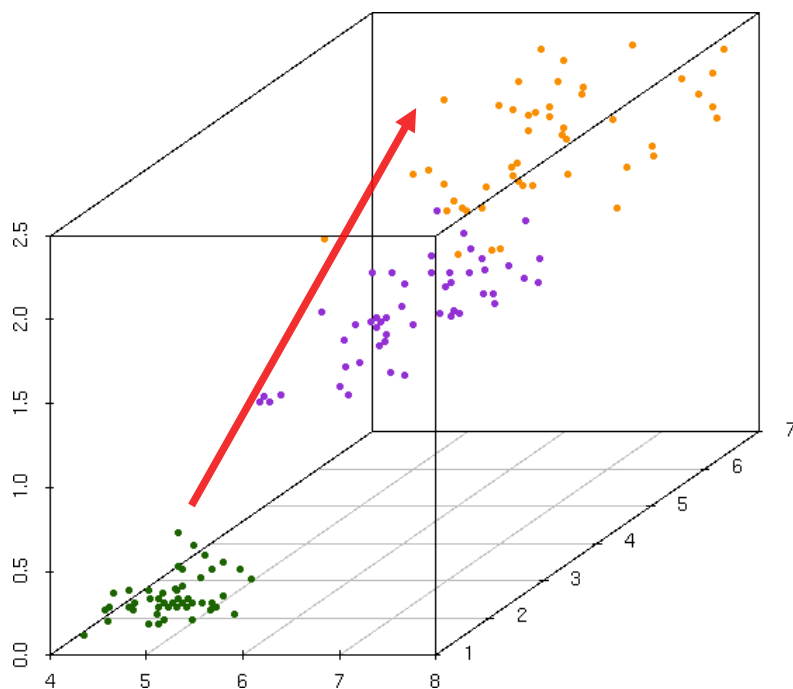
$$\text{Euclidian distance} = \sqrt{(5 - 1)^2 + (4 - 1)^2} = 5$$

Distância Euclidiana

- N-dimensional

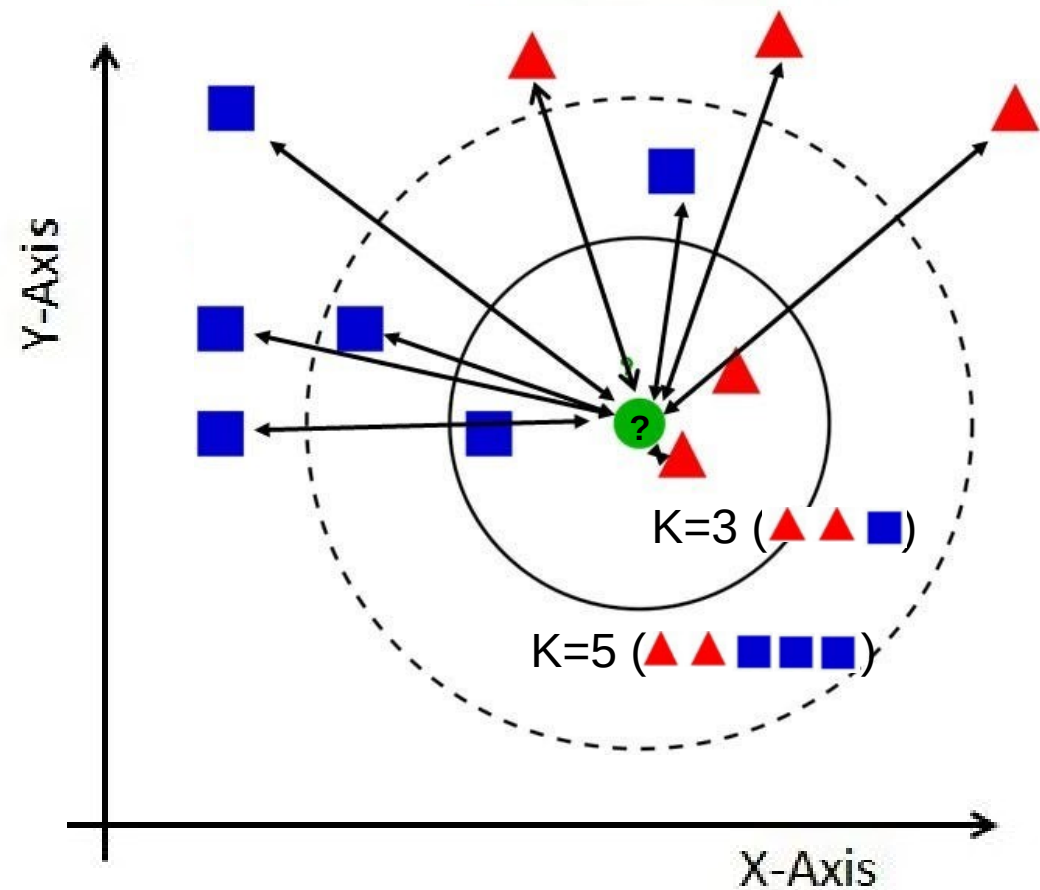
$$d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^n (q_i - p_i)^2}.$$



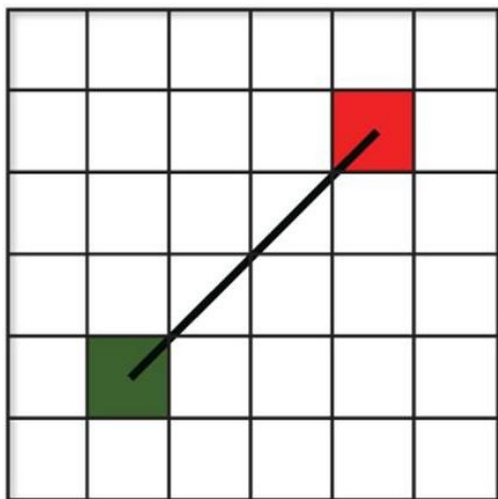
Inferência KNN

- Computar as distâncias entre a amostra de teste e as amostras de treino
- Selecionar os K vizinhos
- Votação

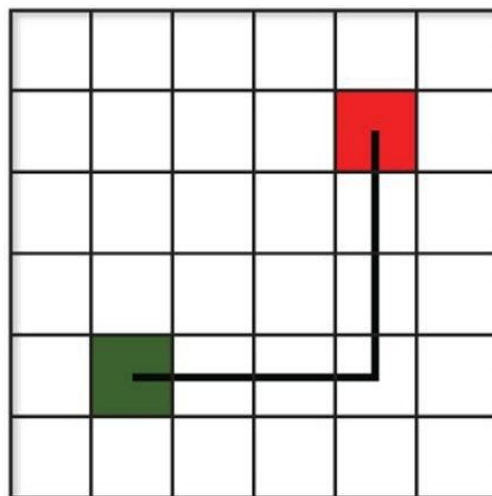


Outras métricas

- Distâncias



Euclidean Distance



Manhattan Distance

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

$$d = \sum_{i=1}^n |x_i - y_i|$$

$$\left(\sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}}$$

Minkowski distance

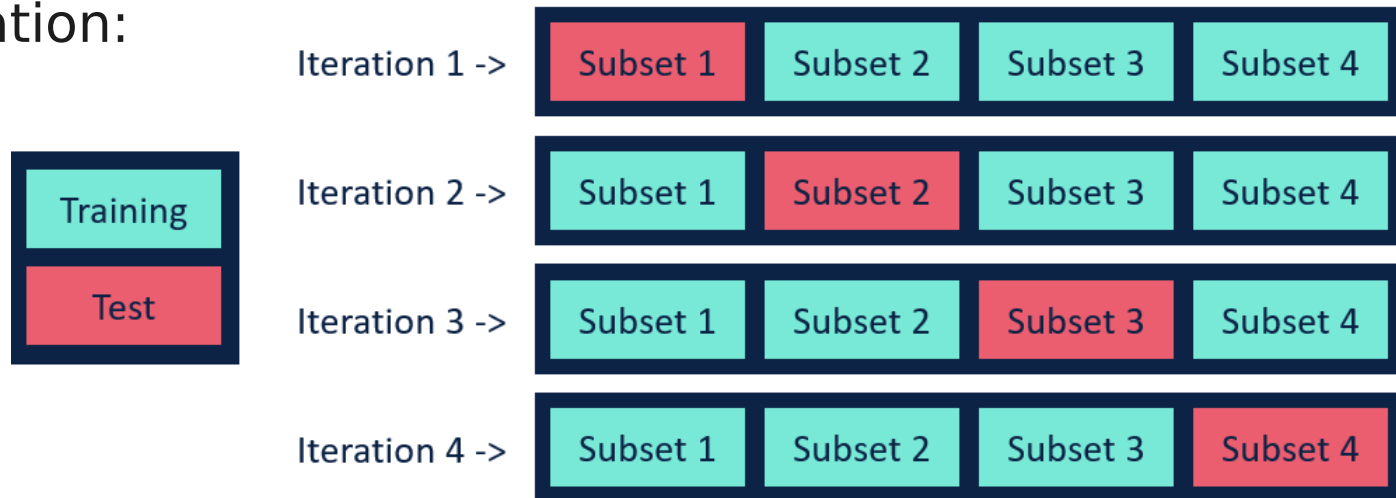
Protocolo Experimental

- Define como um modelo vai ser avaliado
- Define quais dados serão usados para treino, validação e teste
- Técnicas mais comuns:

- Holdout



- Cross-Validation:



Métricas de Desempenho

- Acertos:
 - TP (True Positive):
 - FN (False Negative)
- Erros:
 - FP (False Positive)
 - TN (True Negative)

True positive



False positive



False negative



True negative



Métricas de Desempenho

- Accuracy:

- Instâncias corretamente classificadas** sobre o total de instâncias

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN}$$

- $(55 + 30) / (55 + 5 + 30 + 10) = 0.850$

| | | PREDICTED LABEL | |
|------------|----------|----------------------|---------------------|
| | | NEGATIVE | POSITIVE |
| TRUE LABEL | NEGATIVE | 55 TRUE NEGATIVE | 5 FALSE POSITIVE |
| | POSITIVE | 10 FALSE NEGATIVE | 30 TRUE POSITIVE |

- Qual o problema com *Accuracy*?

- Dados desbalanceados

- Acc: 90% (90/100)
 - Error TP: 100% (10/10)

| | | PREDICTED LABEL | |
|------------|----------|----------------------|---------------------|
| | | NEGATIVE | POSITIVE |
| TRUE LABEL | NEGATIVE | 90 TRUE NEGATIVE | 0 FALSE POSITIVE |
| | POSITIVE | 10 FALSE NEGATIVE | 0 TRUE POSITIVE |

Métricas de Desempenho

- Precisão:
 - **Instâncias positivas classificadas corretamente** sobre o total de instâncias classificadas como positivas

$$Precision = \frac{TP}{TP + FP}$$

$$30/(30 + 5) = 0.857$$

-
- Recall
 - **Instâncias positivas classificadas corretamente** sobre o **total de instâncias positivas** (A.K.A Sensitivity or TP Rate)

$$Recall = \frac{TP}{TP + FN}$$

- $30/(30 + 10) = 0.750$

TRUE LABEL

| PREDICTED LABEL | | | |
|-----------------|----------------------|---------------------|--|
| | NEGATIVE | POSITIVE | |
| NEGATIVE | 55 TRUE NEGATIVE | 5 FALSE POSITIVE | |
| POSITIVE | 10 FALSE NEGATIVE | 30 TRUE POSITIVE | |

Métricas de Desempenho

- F1-SCORE:
 - Média Harmônica^(*) entre precisão e recall

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

- $2 * (0.857 * 0.75) / (0.857 + 0.75) = 0.79$

- Discussão

- Accuracy: 0.850
- F1-Score: 0.799
 - Precision: 0.857
 - Recall: 0.750

(*) A média harmônica atribui menos peso aos valores maiores e mais peso aos valores menores.

| | | PREDICTED LABEL | |
|------------|----------|----------------------|---------------------|
| | | NEGATIVE | POSITIVE |
| TRUE LABEL | NEGATIVE | 55 TRUE NEGATIVE | 5 FALSE POSITIVE |
| | POSITIVE | 10 FALSE NEGATIVE | 30 TRUE POSITIVE |

Let's Code

- Vamos implementar esses conceitos, siga o link:
- [Tópico_02_Aprendizado_Supervisionado_KNN.ipynb](#)

Considerações Finais

- KNN é um método não paramétrico, baseado na vizinhança Euclidiana
- Não tem treinamento
- Desempenho bom em cenários linearmente separáveis
- Tempo é um problema para bases grandes ou altas dimensões